UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/584,167	06/22/2006	Takaharu Hamada	OMOR-0012	5067

23377 7590 10/27/2010
WOODCOCK WASHBURN LLP
CIRA CENTRE, 12TH FLOOR
2929 ARCH STREET
PHILADELPHIA, PA 19104-2891

EXAMINER	
HAYIM, SAMUEL E	

ART UNIT	PAPER NUMBER
2192	

NOTIFICATION DATE	DELIVERY MODE
10/27/2010	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

eofficemonitor@woodcock.com

Office Action Summary	**Application No.** 10/584,167	**Applicant(s)** HAMADA, TAKAHARU	
	Examiner SAMUEL HAYIM	**Art Unit** 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11 August 2010*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-12* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-12* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)	4) □ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)	
3) □ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.	5) □ Notice of Informal Patent Application
	6) □ Other: _____.

Detailed Action

1. Applicant's amendment and response dated August 11th 2010, responding to the May 11th 2010, Office Action provided in the rejection of claims 1-12. Claims 1 and 11-12 have been amended. Claims 1-12 are pending in this application and which have been fully considered by the examiner.

Applicant primarily arguing for the claims not being unpatentable over the *Bryant-Rich* patent in view of the *Roth* patent because neither *Bryant-Rich* nor *Roth* alone or in combination disclose judging if the change is within the standard range that was established at the time of the initial installation qualification (See pages 7-8 of the amendment and response) are not persuasive, as will be addressed under Prior Art's Arguments - Rejections section at item 2 below. Accordingly, the rejection of the claims over the prior art in the previous office action is maintained and **THIS ACTION IS MADE FINAL.** *See* MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply later than SIX MONTHS from the date of this final action.

Prior Art's Arguments – Rejections

2. Applicant's arguments filed on August 11th 2010, in particular pages 7-8, have been fully considered but they are not persuasive. For example:

On page 7-8, Applicant contends that claim 1 is not unpatentable over *Bryant-Rich* in view of the *Roth* as neither *Bryant-Rich* nor *Roth* alone or in combination disclose judging if the change is within the standard range that was established at the time of the initial installation qualification. However, Examiner strongly disagrees. For example, Bryan-Rich paragraph 24 and figure 2, "the purpose of the cleanup service is to perform the undoing, of the change(s) made by the application to the computer's non-volatile memory, that is performed when the computer boots subsequent to an abnormal termination of the application." The standard range that is established upon installation of the program includes all changes made to the application program (and controlled memory regions) however, the range falls short of unexpected errors (resulting in computer crash) that the clean-up service handles. The inspection scenarios monitor state changes such as the ones feature in box 58-60 of figure 2 and system failure. The boxes in the figure 2 indicate states that are within the range (normal operation) while system failure (abnormal operation) is judged outside the range, necessitating the clean-up service.

Double Patenting

3. Examiner acknowledges receipt of the terminal disclaimer filed August 11th 2010, regarding patent 7,349,810. Accordingly, double patenting rejection regarding claims 1-2, 4 and 8-9 is hereby withdrawn.

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1-3, 6-7 and 10-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bryant-Rich (US 2005/0114643) (hereinafter Rich) in view of Roth (US 5,857,205).

As per claim 1, Rich discloses a computer software program article comprising a storage medium having stored thereon a computer software program which when executed causes a detection of whether or not there have been one or more specific changes in one or more application programs running on a computer system, said computer software program comprising: (For example, figure 1-2, launcher, 14, monitors changes made in applications, 14, running on the computer, 30).

an authentication program for authenticating a user prior to the execution of at least one of said application programs, and for associating the user in the aforementioned authentication to an application program that will be run later; (For example, paragraph 3, the use of the computer is restricted to a specified list of users, the applications are installed on the Personal Computer and the configuration and data used by each application are stored separately for each user. A personal computer often also has a generally usable but very restricted access method such as a "guest" account. A program is configured and stored separately from other users thereby associating that particular program with that particular user).

inspection scenarios, associated with each of said application programs and stored on said storage medium, for detecting automatically whether or not there have been specific changes in each of said application programs; (For example, figure 2, box 58 and 60 are examples of two inspection scenarios that monitor state changes, automatically, in the application program being executed or the memory device being removed from the computer).

by judging if the change is within the standard range that was established at the time of the initial installation qualification, during the operation of each of said application programs (For example, paragraph 24 and figure 2, “the purpose of the cleanup service is to perform the undoing, of the change(s) made by the application to the computer's non-volatile memory, that is performed when the computer boots subsequent to an abnormal termination of the application.” The standard range that is established upon installation of the program includes all changes made to the application program (and controlled memory regions) however, the range falls short of unexpected errors (resulting in computer crash) that the clean-up service handles. The inspection scenarios monitor state changes such as the ones feature in box 58-60 of figure 2 and system failure. The boxes in the figure 2 indicate states that are within the range (normal operation) while system failure (abnormal operation) is judged outside the range, necessitating the clean-up service).

and an inspection scenario program for detecting whether or not there has been a specific change in an application program, by running said associated specific application program according to said associated inspection scenario, (For example, figure 1-2, the launcher, 14 is the inspection scenario program as it detects the state and memory changes with the application that it is executing).

Rich does not expressly disclose for outputting detection results in association with said user name and said application program

However, Roth discloses for outputting detection results in association with said user name and said application program (For example, column 1 line 66 to column 2 line 7, in any case, when batch systems encounter undetected errors in the data, the process may or may not respond to the error. In the case where the process is affected by the error, it will either notify the user of a problem in a controlled fashion (if the possibility of that type of error was foreseen) or the process will be forced to a halt (when the error is of an unforeseen nature). The error in the data may also go undetected allowing the process to continue to completion, so that the incorrect data will not be immediately obvious. The errors (changes) output reflect the user associated with the program (under examination).

Rich and Roth are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and output the results of the changes made to the user associated with the program executed as taught by Roth because it would provide for the efficient means for detecting errors and providing data verification throughout the entire computer system (see Roth column 3, line 1-4).

As per claim 2, Rich discloses computer software program article as set forth in claim 1, wherein: said one or more application programs includes an application program executor for displaying the other application programs to the user selectively and executably (For example, figure 3, the launcher, 14, displays a list of applications, 16-16'', to the user for selection).

As per claim 3, Rich does not expressly disclose a computer software program article as set forth in claim 1, wherein: said one or more application programs is a plurality of application programs requiring validations in order to fulfill specific standards under identical policies; said inspection scenarios are for detecting whether or not changes in each application program are to the degree that allows execution of the application program without performing a re-validation; and said inspection scenario program instructs said computer system to display the detection result, when, in accordance with said inspection scenario, a change of an extent greater than that wherein execution is allowed without performing validations in detected.

However, Roth discloses wherein: said one or more application programs is a plurality of application programs requiring validations in order to fulfill specific standards under identical policies; (For example, figure 1, files (from multiple programs) are validated to identical standards using file analysis, 14 during process monitoring (execution), 16).

said inspection scenarios are for detecting whether or not changes in each application program are to the degree that allows execution of the application program without performing a re-validation; (For example, figure 1, data verification and validation is based on errors (changes) in the files accessed by the applications. Serious anomalies and significant variations affect the ability for the application to run; For example, column 1 line66 to column 2 line 7, In any case, when batch systems encounter undetected errors in the data, the process may or may not respond to the error. In the case where the process is affected by the error, it will either notify the user of a problem in a controlled fashion (if the possibility of that type of error was foreseen) or the process will be forced to a halt (when the error is of an unforeseen nature). The error in the data may also go undetected allowing the process to continue to completion, so that the incorrect data will not be immediately obvious).

and said inspection scenario program instructs said computer system to display the detection result, when, in accordance with said inspection scenario, a change of an extent greater than that wherein execution is allowed without performing validations in detected (For example, figure 1 and column 34 line 29-36, employs a generic approach which is driven by record descriptions (a.k.a. record layouts) which may be created for use in programs which read from or write to these files. This software may be used to profile the contents of files, monitor changes, detect likely areas of erroneous data, generate data domain meta-data, and verify "migrated" information in parallel implementations and similar uses. Figure 1 depicts a series of generated reports that report on errors (changes) whose threshold is set to any. Therefore, the threshold is immediately passed at the occurrence of the first error, generating the reports).

Rich and Roth are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and output the results of the changes made to the user associated with the program executed as taught by Roth because it would provide for the efficient means for detecting errors and providing data verification throughout the entire computer system (see Roth column 3, line 1-4).

As per claim 6, Rich does not expressly disclose a computer software program article as set forth in claim 1, wherein: said inspection scenario program is launched as specific time intervals

However, Roth discloses wherein: said inspection scenario program is launched as specific time intervals (For example, column 4 line 42-49, the first part compares record layouts over time to determine if they have changed in ways that would affect the contents of files. The second part performs a generic data item evaluation that obtains a description of the contents of every data item that is identified in the record layouts (a.k.a. data item characteristics), and compares these characteristics over time (where historical information is available). The comparison of file layouts over time occurs over a specific time intervals as random does not exist within computer architectures. Therefore, a specific time interval is set).

Rich and Roth are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and output the results of the changes made to the user associated with the program executed as taught by Roth because it would provide for the efficient means for detecting errors and providing data verification throughout the entire computer system (see Roth column 3, line 1-4).

As per claim 7, Rich does not expressly disclose a computer software program article as set forth in claim 1, wherein: said inspection scenario program comprises a detection results display step for displaying said detection results on a computer display, and a user input/output step for receiving user input regarding said detection results and for outputting in association with said detection results.

However, Roth discloses wherein: said inspection scenario program comprises a detection results display step for displaying said detection results on a computer display, and a user input/output step for receiving user input regarding said detection results and for outputting in association with said detection results (For example, figure 3C, the detection results are displayed to the user. The user is able to manipulate the files; For example, column 6 line 44-65, The first group of items which the online entry procedure prompts the user for are the "File Identification" items 24 such as the "File Name (Descriptive Name)" and "DSN (Use "0" for GDGs)." The "File Name (Descriptive Name)" is the name of the file in plain language. It serves as an essential piece of system documentation. The "DSN (Use "0" for GDGs)" is the "data set name" and is the "formal" name used by the method to "catalogue" the file. The "File ID" that is subsequently assigned to each file reference is based primarily on the DSN. The reason for substituting a numerical key in place of the DSN is mainly as a space and time saving measure.

A DSN (on the MVS system) can be 44 bytes long, the binary packed numerical file ID occupies only 2 bytes. Another reason for using a File ID alias involves the situation where a file's DSN has to be changed. In such a situation, the File ID can be reassigned independently of the DSN, thus maintaining the continuity of references across file generations. If the file is a generation data group (GDG) the user will follow the DSN with "(0)" to indicate the current version. Entering the DSN of an already specified file entry will cause that entry to be retrieved for maintenance purposes. Files are therefore selected based on the users input).

Rich and Roth are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and output the results of the changes made to the user associated with the program executed as taught by Roth because it would provide for the efficient means for detecting errors and providing data verification throughout the entire computer system (see Roth column 3, line 1-4).

As per claim 10, Rich does not expressly disclose a computer software program article as set forth in claim 1, wherein said computer software program is for insuring that the results of safety testing have not been falsified or tampered with in said application program; and wherein said application program receives measurement values, which have not been falsified or tampered with, from a measurement device for safety testing, processes the measurement values, and outputs the results of specific processing.

However, Roth discloses wherein said computer software program is for insuring that the results of safety testing have not been falsified or tampered with in said application program; (For example, column 4 line 10-27, file records are check against baseline versions of the same file thereby insuring that the files have not been tampered or falsified regardless of what the files are for).

and wherein said application program receives measurement values, which have not been falsified or tampered with, from a measurement device for safety testing, processes the measurement values, and outputs the results of specific processing (For example, figure 1 column 3 line 42-49, the data is aggregated for a repository of files to be checked (layouts comparison confirms tamper proofed files). The system is a safety testing system as file integrity is safety checking. The files are processed at, 14; For example, figure 3C, the results are output to the user).

Rich and Roth are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and output the results of the changes made to the user associated with the program executed as taught by Roth because it would provide for the efficient means for detecting errors and providing data verification throughout the entire computer system (see Roth column 3, line 1-4).

As per claim 11, Rich discloses an application program inspecting system comprising: an application storage unit for storing one or more mutually-related application programs; (For example, figure 1, application programs are stored in memory 12).

an authentication unit for performing user authentication prior to running at least the first of said application programs, and for associating application programs run thereafter, with the user involved in said authentication; (For example, paragraph 3, the use of the computer is restricted to a specified list of users, the applications are installed on the Personal Computer and the configuration and data used by each application are stored separately for each user. A personal computer often also has a generally usable but very restricted access method such as a "guest" account. A program is configured and stored separately from other users thereby associating that particular program with that particular user).

an inspection scenario storage unit for storing inspection scenarios, associated with each of said application programs, for detecting automatically whether or not there have been specific changes in each of said application programs; (For example, figure 2, box 58 and 60 are examples of two inspection scenarios that monitor state changes, automatically, in the application program being executed or the memory device being removed from the computer).

by judging if the change is within the standard range that was established at the time of the initial installation qualification, during the operation of each of said application programs (For example, paragraph 24 and figure 2, “the purpose of the cleanup service is to perform the undoing, of the change(s) made by the application to the computer's non-volatile memory, that is performed when the computer boots subsequent to an abnormal termination of the application.” The standard range that is established upon installation of the program includes all changes made to the application program (and controlled memory regions) however, the range falls short of unexpected errors (resulting in computer crash) that the clean-up service handles. The inspection scenarios monitor state changes such as the ones feature in box 58-60 of figure 2 and system failure. The boxes in the figure 2 indicate states that are within the range (normal operation) while system failure (abnormal operation) is judged outside the range, necessitating the clean-up service).

and an inspecting unit for detecting whether or not there have been specific changes in said application programs, through an inspection scenario program executing specific related application programs in accordance with said associated inspection scenarios, (For example, figure 1-2, the launcher, 14 (inspection scenario program) executes the specific related application program and determines changes made to the application programs, 16, using the associated scenarios).

Rich does not expressly disclose and for outputting said detection results in association with said user name and application program.

However, Roth discloses and for outputting said detection results in association with said user name and application program (For example, figure 3C, the output file contains the name of the application program (top left written as 'XXXXXXX') and information of the user as seen in lines 11-20 which contain information regarding the name, birthday, hired time, sex, job type, job title, and job class).

Rich and Roth are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and output the results of the changes made to the user associated with the program executed as taught by Roth because it would provide for the efficient means for detecting errors and providing data verification throughout the entire computer system (see Roth column 3, line 1-4).

As per claim 12, Rich discloses a method implemented using a computer for detecting whether or not there have been one or more specific changes in one or more mutually-related application programs on a computer system, (For example, figure 1-2, applications are executed on host computer, 30, where changes are detected relating to specific programs).

comprising: an authentication process for performing user authentication prior to running at least the first of said application programs and for associating, with said application programs run thereafter, the user involved in said authentication; (For example, paragraph 3, the use of the computer is restricted to a specified list of users, the applications are installed on the Personal Computer and the configuration and data used by each application are stored separately for each user. A personal computer often also has a generally usable but very restricted access method such as a "guest" account. A program is configured and stored separately from other users thereby associating that particular program with that particular user).

and an inspection process for detecting whether or not there has been a specific change in said application program, through the use of an inspection scenario, associated with each application program, for detecting, automatically whether or not there has been a specific change in each of the application programs, by judging if the change is within the standard range that was established at the time of the initial installation qualification, during the operation of each of said application programs by an inspection scenario program executing a specific related application program in accordance with said associated inspection scenario, (For example, figure 1-2, the system comprises a series of inspection scenarios, boxes 58-60 and 64, which automatically detect specific changes to the application program. The inspection scenarios are stored in memory (figure 1); For example, paragraph 24 and figure 2, "the purpose of the cleanup service is to perform the undoing, of the change(s) made by the application to the computer's non-volatile memory, that is performed when the computer boots subsequent to an abnormal termination of the application." The standard range that is established upon installation of the program includes all changes made to the application program (and controlled memory regions) however, the range falls short of unexpected errors (resulting in computer crash) that the clean-up service handles. The inspection scenarios monitor state changes such as the ones feature in box 58-60 of figure 2 and system failure. The boxes in the figure 2 indicate states that are within the range (normal operation) while system failure (abnormal operation) is judged outside the range, necessitating the clean-up service; For example, figure 1-2, the launcher, 14 (inspection scenario program) executes the specific related application program, 16, and determines changes made to the application programs using the associated scenarios).

Rich does not expressly disclose and outputting the detection results in association with said user name and application program.

However, Roth discloses and outputting the detection results in association with said user name and application program (For example, figure 3C, the output file contains the name of the application program (top left written as 'XXXXXXX') and information of the user as seen in lines 11-20 which contain information regarding the name, birthday, hired time, sex, job type, job title, and job class).

Rich and Roth are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and output the results of the changes made to the user associated with the program executed as taught by Roth because it would provide for the efficient means for detecting errors and providing data verification throughout the entire computer system (see Roth column 3, line 1-4).

3. Claims 4-5 and 8-9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bryant-Rich (US 2005/0114643) (hereinafter Rich) in view of Roth (US 5,857,205) in further view of Ribot (US 2003/0187993).

As per claim 4, Rich does not expressly disclose a computer software program article as set forth in claim 1, wherein: said inspection scenario program inputs a dummy signal into said application program in accordance with said inspection scenario to detect a response signal to said inputted dummy signal, to thereby detect whether or not there has been a specific change in the application program.

However, Ribot discloses wherein: said inspection scenario program inputs a dummy signal into said application program in accordance with said inspection scenario to detect a response signal to said inputted dummy signal, to thereby detect whether or not there has been a specific change in the application program (For example, paragraph 41-49, the distributed object CommMgr in, accordance with the present invention inherits from a second object whose only function is to limit the range of operations of the distributed object CommMgr. This second object does not define any operation (or method), but contains the reference of an organization and/or of a BN 4, which specifies the domain of application of the operations, i.e. it is a data object. This second object is resident in server 11 or could be provided on another node of the OMN 16, e.g. in database server 19. Each proxy (CommMgr) distributed by the MD 10 defines exactly the domain of application of offered services. These data cannot be modified by the client organization as the client's terminals have read-only access to the attributes defining the client's profile. So, at the time the client organization requests a first connection, the receiving server of the OMN 16, after authentication, will generate the proxy to be sent to that client organization in function of the services and domains to which the organization has rights. This proxy is now available at the client organization's external terminal. The next request from the client's terminal will access the proxy which enables a comparison of the request and the authorization credentials in the proxy. Depending upon the comparison, the proxy can enable forwarding of the request. For example, if the request and the privileges do not match, the request is aborted and/or any other suitable action is taken to prevent the request being made. On the other hand if the authorization credentials are consistent with the request, the proxy enables forwarding of the request to the server 16. Afterwards, it will not be necessary to verify the rights of the client organization as the fact that the client organization is in possession of a proxy which has allowed the request to proceed to the server will be sufficient for proving the rights of the client organization. The two proxies are added to the interchange between client and server to protect the integrity of the network resource. The proxies thereby refer to a dummy signal that is used to detect changes (requests for changes to network resources)).

Rich and Ribot are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and control access between the user and the software as taught by Ribot because it would provide for the efficient means for controlling security and access controls during the authentication and authorization of the client organization (see Ribot paragraph 35).

Rich does not expressly disclose and associates, with the user involved in said authentication, the application programs currently running, and application programs run thereafter.

As per claim 5, Rich discloses a computer software program article as set forth in claim 4, wherein: said inspection scenario includes at least data for specifying an application program to be subjected to inspection, (For example, figure 2, the launcher is executed, 54, (control program to detect changes made to application program) then the application specified is executed at box 56).

and data regarding an allowable range regarding the response to said data (For example, figure 2, the responses from the application as it is executed is detected by launcher (boxes 58-60 and 64) that determine the allowable range of changes made to the application).

Rich does not expressly disclose data for inputting as dummy signals into said application program.

However Ribot discloses data for inputting as dummy signals into said application program, (For example, paragraph 41-49, the distributed object CommMgr in, accordance with the present invention inherits from a second object whose only function is to limit the range of operations of the distributed object CommMgr. This second object does not define any operation (or method), but contains the reference of an organization and/or of a BN 4, which specifies the domain of application of the operations, i.e. it is a data object. This second object is resident in server 11 or could be provided on another node of the OMN 16, e.g. in database server 19. Each proxy (CommMgr) distributed by the MD 10 defines exactly the domain of application of offered services. These data cannot be modified by the client organization as the client's terminals have read-only access to the attributes defining the client's profile. So, at the time the client organization requests a first connection, the receiving server of the OMN 16, after authentication, will generate the proxy to be sent to that client organization in function of the services and domains to which the organization has rights. This proxy is now available at the client organization's external terminal. The next request from the client's terminal will access the proxy which enables a comparison of the request and the authorization credentials in the proxy. Depending upon the comparison, the proxy can enable forwarding of the request. For example, if the request and the privileges do not match, the request is aborted and/or any other suitable action is taken to prevent the request being made. On the other hand if the authorization credentials are consistent with the request, the proxy enables forwarding of the request to the server 16. Afterwards, it will not be necessary to verify the rights of the client organization as the fact that the client organization is in possession of a proxy which has allowed the request to proceed to the server will be sufficient for proving the rights of the client organization. The two proxies are added to the interchange between client and server to protect the integrity of the network resource. The proxies thereby refer to a dummy signal that is used to detect changes (requests for changes to network resources) and data exists within the system to build the proxies between the client and server).

Rich and Ribot are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and control access between the user and the software as taught by Ribot because it would provide for the efficient means for controlling security and access controls during the authentication and authorization of the client organization (see Ribot paragraph 35).

As per claim 8, Rich does not expressly disclose a computer software program article as set forth in claim 1, wherein said authentication program: comprises an authentication update requesting article for requesting the input of user authentication data at each specific time interval; and if the user cannot be authenticated by said authentication update requesting article, terminates said application program that is running, associated with the applicable user.

However, Ribot discloses wherein said authentication program: comprises an authentication update requesting article for requesting the input of user authentication data at each specific time interval; (For example, paragraph 49, the user authentication occurs each time a request is made to the network resource. Each request is a specific time interval).

and if the user cannot be authenticated by said authentication update requesting article, terminates said application program that is running, associated with the applicable user (For example, paragraph 50, the access control decision function (ACDF) (which is the procedure (or set of procedures) that applies the access control rules to each access request so as to determine whether the requested access to a management object should be granted or denied) is implemented by the proxy in accordance with the present invention. The access control enforcement function (ACEF) (which is the procedure (or set of procedures) for enforcing the decisions made by the ACDF) is also performed by the proxy. Authorization is denied when a network resource (management object), or application, is accessed by an unauthorized user thereby halting the application).

Rich and Ribot are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and control access between the user and the software as taught by Ribot because it would provide for the efficient means for controlling security and access controls during the authentication and authorization of the client organization (see Ribot paragraph 35).

As per claim 9, Rich does not expressly disclose a computer software program article as set forth in claim 1, wherein: said authentication program performs repeat user authentication in response to a user request after an initial user authentication.

However, Ribot discloses a computer software program article as set forth in claim 1, wherein: said authentication program performs repeat user authentication in response to a user request after an initial user authentication, (For example, paragraph 12-13, the distributed proxy communications object component is adapted to enable comparison of the contents of the request and the definition of the rights and privileges of the user. The request relates to modification of a management object maintained at a network resource, the organization having a global right to access the network resource. A second client proxy communications object component is preferably adapted to enable forwarding of the request to the server in response to the comparison. Preferably, the request is forwarded only when comparison step determines that the request and the rights and privileges are consistent, Thereby controlling individual authorization of the user each time network resource is accessed).

Rich and Ribot are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and control access between the user and the software as taught by Ribot because it would provide for the efficient means for controlling security and access controls during the authentication and authorization of the client organization (see Ribot paragraph 35).

Rich does not expressly disclose and associates, with the user involved in said authentication, the application programs currently running, and application programs run thereafter.

However, Roth disclose and associates, with the user involved in said authentication, the application programs currently running, and application programs run thereafter (For example, column 1 line 66 to column 2 line 7, in any case, when batch systems encounter undetected errors in the data, the process may or may not respond to the error. In the case where the process is affected by the error, it will either notify the user of a problem in a controlled fashion (if the possibility of that type of error was foreseen) or the process will be forced to a halt (when the error is of an unforeseen nature). The error in the data may also go undetected allowing the process to continue to completion, so that the incorrect data will not be immediately obvious. The errors (changes) output reflect the user associated with the program (under examination).

Rich and Roth are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and output the results of the changes made to the user associated with the program executed as taught by Roth because it would provide for the efficient means for detecting errors and providing data verification throughout the entire computer system (see Roth column 3, line 1-4).

Conclusion

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Samuel Hayim whose telephone number is (571) 270-3370. The examiner can normally be reached on Monday to Friday 8:30 AM to 5:00 PM.

If attempts to reach the above noted Examiner by telephone are unsuccessful, the Examiner's supervisor, Tuan Dam, can be reached at the following telephone number: (571) 272-3695.

The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/SAMUEL HAYIM/
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192